APPENDIX C

1. PIM-FILTER

Appendix C will be understood and appreciated from the following detailed description, taken in conjunction with the drawings in which:

Fig. 57 is an example of a BGA (ball grid array) board to be inspected by a system constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 58 is a close-up view of the lower end of the BGA board of Fig. 57;

Fig. 59 is a close-up view of a bond finger on a BGA board, constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 60A is a flow chart illustrating data flow through a PIM map generator during the Learn stage of an inspection scan, constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 60B is a flow chart illustrating data flow through a filter during the Inspect stage of an inspection scan, constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 61 is a simplified block diagram illustrating subtraction of regions by the map generator, constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 62 is a simplified block diagram illustrating the data structure of a PIM stored in runlength format, constructed and operative in accordance with a preferred embodiment of the present invention; and

Fig. 63 is a simplified block diagram of the general structure of the PIM system, constructed and operative in accordance with a preferred embodiment of the present invention.

1. Introduction

The scanned area of an inspected object, such as a BGA panel, contains regions of varying importance, ranging from sensitive bonding regions and up to text regions that are to be ignored. The Image Processing procedure should treat each region with different criteria of sensitivity in order to obtain maximal detection and minimal false alarms. Two concepts of differentiating between areas are possible: filtering out information in non-interesting regions, and region-dependent type and sensitivity of inspection.

In accordance with a preferred embodiment of the present invention, image processing is composed of several phases: defect detection, defect verification, and top-down windows. The defect detectors are realized both in hardware, like the nick-protrusion and surface defects channels, and in software, like the excess-missing and line-width channels. Each of these detectors produces suspicious events that might, or might notsignal, real defects; some of events may be certain defects and preferably need to be verified by a more careful inspection such as by CEL-to-CEL analysis as described with reference to Figs. 30 - 30C in the specification. Another method of detecting suspicious events is by preparing in advance (during the Learn scan) top-down windows in areas of special interest, and examining them more carefully during Inspect.

In this appendix, two, apparently different, methods that realize region differentiating are discussed:

- Filter - screens events according to their type and position, ignoring irrelevant areas such as text areas, areas known as "unstable", etc. Placing the filter at a critical position in the IP pipeline would also save slow processing of non-interesting areas and increase speed and efficiency.
- Tolerance Setting - specifies the accuracy of inspection test of events, according to their type and geometric location with respect to regions of interest.

Both these methods rely on the same geometric query tool, namely Point In Map (PIM), that answers the following question: which region in a map contains a point (event).

FIGURE 57. is an example of a BGA board. The thick black regions are exposed metal and the stippled regions and thin line regions are metal covered with solder mask. The exposed metal regions are important and preferably are inspected carefully, while covered areas, typically, are less critical.

2.1 Filter

2.1.1 Scope

The Filter, which will be realized as a SIP task, receives as input a list of events and produces a new list that contains only the "interesting" events. The filtering criteria

are geometry and type, and can be either set by the user or automatically constructed during the Learn process.

Filtering is performed in several levels: filtering of raw data, namely CELs and features, filtering of events suspected as defects resulting from the defect detectors, and filtering of certain defects resulting from the verifiers.

Filtering raw data has two main drawbacks: it can be time consuming since it is realized in software, and in the case of CELs it generates "holes" in the incoming data, which can be harmful to other processes such as the connected components application. Filtering of suspected events is less damaging to the picture and reduces substantially the amount of features for filtering, however it compels redundant processing of ignoring areas by the defect detectors. Filtering of verified defects compels verification, which is a long and difficult process, of excessive events and thus is inferior to the other two.

The Filter uses the PIM geometric tool for map generation and point location queries. The format of input points is scan line format, ordered lines of events in increasing y-coordinate. This can be used to increase the efficiency of search by using the knowledge of the former point. More details are discussed in the PIM description in section 3.3.

FIGURE 58. is a close up of the lower end of the BGA picture, with a region to be ignored being marked.. The marked region is misalignment between the solder mask and the metal which is not a defect, yet might cause false calls.

2.1.2 Input

The input of the filter task includes the following features:

- filtering map - a planar subdivision of the scan area is generated by the PIM according to the filtering regions, where every point (x,y) is contained inside a single face. Each face represents a different filtering criterion, namely type(s). The map is given in "ideal" reference coordinates, meaning after adjustments of pixel size, shape, etc. If the scan area is composed of several duplicates of the same pattern, namely step and repeat, only a single-repeat map is needed.
- registration - an alignment transformation between the ideal Learn coordinate system (filtering map) and the ideal Inspect system (on-line features). If no registration

is given, as in the case of design-rule defects computed during the learn inspection process, the unit transformation is taken. The allowed transformation is affine, namely shift, rotation, and expansion.

- events - a list of events in scan line format, characterized by type and position.

2.1.3 Output

- events - a partial group of the Input list of events in on-line coordinates, that contains only the ones that are not filtered.

2.2 Tolerance Setting

2.2.1 Scope

The verification test function performs an accurate measurement of "suspected" defects and decides, according to previously defined sensitivity criteria, whether it is a real defect or not. The sensitivity criteria depend on the exact location of the defect. Thus a geometric location tool is needed to determine the location of the defect and hence the accuracy of measurement necessary for verification.

The Tolerance Setting is a service function (func_defects_handler) that is used either before or after the actual test function is applied. If the type of test depends on the accuracy of measurement, this service is called before the test in order to determine its type. If the type of test does not depend on the allowed tolerance, this service is used for sensitivity setting. It should be noted that one of the test functions possible is filtering. In this context filtering is performed for arbitrary positioned events, hence falling outside the Filter scope.

The Tolerance Setting function asks the same geometric location queries as the Filter, however the format of input events is position random instead of ordered. The PIM tool gives services of map generation and point queries also to the Tolerance Setting application.

FIGURE 59 is a close up of a bond finger as an example of varying inspection sensitivity. Region 1 is unstable, meaning its exact location is may change, because of the edge of the solder mask. Region 2 is a critical bonding region. Region 3 is non-critical. Hence region 2 is the most sensitive, and then in descending order regions

3 and 1.

2.2.2 Input

• sensitivity map - a planar subdivision of the scan area generated by the PIM according to interest regions, where every point (x,y) is contained inside a single face. Each face represents a different defect sensitivity, namely allowed tolerance, represented by some type. The map can cover only part of the scan area, such as a trigger window or a single repeat in step and repeat panels. The map is given in ideal reference coordinates.

• registration - an alignment transformation between the ideal Learn coordinate system (sensitivity map) and the ideal Inspect system (on-line features). If no registration is given, as in the case of design-rule defects computed during the learn inspection process, the unit transformation will be taken. The allowed transformation is affine, namely shift, rotation, and expansion.

• event - "suspected" defect or trigger for inspection, in arbitrary position, with type of test function to be applied.

• mask table - a Look Up Table (LUT) that associates the map types to a different inspection sensitivity.

2.2.3 Output

• inspection accuracy - sensitivity of inspection, such as maximal allowed nick size, that will determines whether a "suspected" defect is real or false.

Design

3.1 Filter

The filter includes the following operations:

1. Take a row or a set of rows of events.

2. Transform the events via registration to the Learn coordinate system. Every several rows the transformation is updated by a new dynamic registration.

3. Send each event and the filtering map to the PIM for a position and type query.

4. Receive from the PIM a positive answer for a position and type match between the event and the map, and a negative answer otherwise.

5. Filter out positive events.

6. Inverse transform the remaining events back to the on-line coordinate system.
7. Generate a data source containing the events that are not filtered.

3.2 Tolerance Setting

Outline of the tolerance setting service:

1. Receives a trigger, or a "suspected" defect + type of defect.
2. Apply registration to convert to Learn coordinates.
3. Send the trigger and the sensitivity map to the PIM for a position query.
4. Receive from the PIM a type denoting the face that contains the trigger.
5. Return to on-line coordinate system.
6. Match an inspection tolerance to the trigger+type according to the mask table.

PIM - Point In Map

3.3.1 Scope

The PIM is a geometric tool designed for two distinct functions: generate a unified map from a set of regions, and locate a point in that map. The Map Generator creates a map, where each face is associated with a set of types. The Location Query determines which face contains the input point and either returns its type(s), or determines whether it matches the point's type and returns a boolean answer.

Editing and Display of the map are not in the scope of the PIM, and are the responsibility of the User Interface Application. Nevertheless, separate tools that support those functions will be developed for simulation purposes.

The two functions of the PIM are demonstrated in the following data flow sketch:

FIGURES 60A - 60B are Examples of data flow during an inspection scan: (60A) map generator during Learn. (60B) Filtering during Inspect.

3.3.2 Map Generator

3.3.2.1 Input

The input for the map generator is a set of regions of general geometric shape, each identified by a single type or a union of types. The regions may be user defined or be learned automatically during the Learn scan according to areas of interest. The PIM accepts regions with the following properties:

- Three possible formats: polygons (a set of points), circles, and run-length.
- Closed and simply connected (for polygons only).
- Overlap is allowed. The type set of the overlap area is determined by the user, and can be the union of the original overlapping type sets, one of the original sets, or any other combination.
- Subtraction of regions is allowed. These "negative" regions are regions with no type, usually defined inside regular regions in order to simplify generation of complex areas. In that case the order of definition is important, namely the latest defined region overrides the former. The negative region is identified by the type to be subtracted from the former defined positive regions. The subtraction operation can also be determined by the user: difference of type sets, symmetric difference, etc.

FIGURE 61. (a) is an example of filtering regions. (b) Importance of definition order: negative region #2 overrides positive region #1, while #3 overrides #2.

3.3.2.2 Output

The output is a unified map of all the regions, resulting in a planar subdivision of the area where every point (x,y) is contained inside a single face. Each face represents a different region or an intersection of regions, and is associated with either a single type or a union of types. The map can cover the whole scan area, or just a part such as a single repeat or a trigger window. The format of the output picture should optimize the storage space, difficulty of realization, and efficiency of location queries for the different input point formats. The first excludes raster and encourages some compressed format. The second supports run-length format, namely a format in which a run of pixels having the same values are represented in a compressed form, rather than a polygon format, since the definition of regions may come in both formats and converting a run-length picture to a set of polygons is significantly harder than the opposite. The third, efficiency of location queries, also supports the run-length option over the polygon one. To conclude, it seems that run-length is the preferred format for the output picture.

3.3.3 Location Query

3.3.3.1 Input

• unified regions map - a planar subdivision of the scan area, each face associated with type(s), in run-length format.

• event - a point associated with a type.

3.3.3.2 Output

There are two types of location queries that produce different output:

1. boolean answer to a location and type match question.
2. set of types that are associated with the face containing the point.

3.3.3.3 Point Location Methods

Point location methods vary according to the exact format of the input points. Clearly, efficient search of an arbitrary point and an ordered set of points are different. First, an arbitrary point search is supplied, and a more efficient method for rows of data is added if necessary.

The fastest and most convenient location of a point can be performed by converting the regions map to a raster format, thereby reducing the query to a random access operation. The main drawback of course resides in the size of such a picture and memory access handling. Next is binary search in a run-length picture which is slower, of the order of log(n) where n is the length of the run-length average row, however it is much more efficient in terms of picture size. If the complexity of pictures is high and the speed of the binary search is not enough, another option is available: addition of an indexing system (Indexing of Ordered Data: SRS) to the run-length picture, enabling random access to small "zoom" areas of the run-length. This system enlarges the stored picture by the additional indexes, however reducing the average row length and hence log(n).

Data structure of PIM

The PIM is stored in runlength format, illustrated in Figure 62.

Additional tools

The PIM is a generic geometrical tool. However, the SIP has specific requirements such as configuration files handling, interpretation of region types, etc.

For that purpose some additional tools are needed as detailed in the next sections. The general structure of the whole PIM system is illustrated in Figure 63.

3.4.1 PIM SIP

PIM SIP contains a PIM and has 2 additional features:

- a translation table from string-types to integer-types.
- a self building method from a regions' file.

The translation table is in order to keep the efficiency of integer-type PIM while providing the user with string-type, meaningful, region identification. The configuration file produced by the user is written in terms of regions' names, and the PIM SIP translates automatically to internal integer-types.

The self reading method provides the other SIP applications an easy interface for constructing a PIM. If the proper file exists (see section), no knowledge of PIM building methods is needed and the PIM SIP will read the file and build itself.

3.4.2 Query Table

A translation table of regions' type to detailed regions' information. A query table is attached to a PIM, and translates its id's (of integer type in case of PIM SIP) to objects that hold quantities related to that region. For example in the case of nick threshold values: maximal allowed width and length. One PIM can be attached to many query tables, each representing a different aspect of interpretation of types: nick/protrusion thresholds, width defects, etc. Query table is templated to the data-objects, that can be one of the existing objects (NickProt, LWdefect, ExcessDif) or user defined.

3.4.3 "MIM"

Manager of a system of PIM and attached query tables. Main function is loading configuration files and construction of PIM plus query tables. Query methods also exist for extracting information of point in map, and more detailed regions' information.

3.4.4 Shape File Handling

A special shape file format, with file reader and file writer to support it, is provided for the purpose of transferring shape information between learn and inspect scans. The file reader/writer uses the double-type shapes of the SIP (Dpolyline, Drect, Dcircle). A Shape_base class, which all shapes inherit from, was also added. Shape_base inherits from Base_factory so that all shapes can be cloned when read from file.

File handling is a separate unit that can be replaced easily by a different format file handling, providing similar methods. Methods include open/clear, read/write, and shape information.